On optimal language compression for sets in PSPACE/poly[*]

N. V. Vinodchandran [†] Marius Zimand [‡]

Abstract

We show that if DTIME$[2^{O(n)}]$ is not included in DSPACE$[2^{o(n)}]$, then, for every set B in PSPACE/poly, all strings x in B of length n can be represented by a string $compressed(x)$ of length at most $\log(|B^{=n}|) + O(\log n)$, such that a polynomial-time algorithm, given $compressed(x)$, can distinguish x from all the other strings in $B^{=n}$. Modulo the $O(\log n)$ additive term, this achieves the information-theoretic optimum for string compression. We also observe that optimal compression is not possible for sets more complex than PSPACE/poly because for any time-constructible superpolynomial function t, there is a set A computable in space $t(n)$ such that at least one string x of length n requires $compressed(x)$ to be of length $2\log(|A^{=n}|)$.

Keywords: compression, time-bounded Kolmogorov complexity, pseudo-random generator.

1 Introduction

In many practical and theoretical applications in computer science, it is important to represent information in a compressed way. If an application handles strings x from a finite set B, it is desirable to represent every x by another shorter string $compressed(x)$ such that $compressed(x)$ describes unambigously the initial x. Minimizing the length of $compressed(x)$ is one of the main goals of this task and ideally one would like to achieve the information-theoretic bound $|compressed(x)| \leq \log(|B|)$, for all $x \in B$. If a set B is computably enumerable, a fundamental result in Kolmogorov complexity states that for all $x \in B^{=n}$, $C(x) \leq \log(|B^{=n}|) + O(\log n)$, where $C(x)$ is the Kolmogorov complexity of x,

[*]Previous versions of this work have been presented at FCT'2011 and CCR'2012.

[†]Department of Computer Science and Engineering, University of Nebraska-Lincoln. Part of the work was done while at Dept. of Computer Science, Johns Hopkins University. This work is supported in part by NSF grant CCF 0916525.

[‡]Department of Computer and Information Sciences, Towson University, Baltimore, MD.; email: mzimand@towson.edu; http://triton.towson.edu/∼mzimand. This work is supported in part by NSF grant CCF 1016158.

i.e., the shortest effective description of x ($B^{=n}$ is the set of strings of length n in B). The result holds because x can be described by its rank in the enumeration of $B^{=n}$. However, in many applications, it is desirable that the unambiguous description is not merely effective, but also efficient. This leads to the general idea of considering time-bounded versions of Kolmogorov complexity. An interesting line of research [Sip83, BFL01, BLvM05, LR05] in time-bounded Kolmogorov complexity, which we also pursue in this paper, focuses on a notion called *time-bounded distinguishing Kolmogorov complexity*, $\mathrm{CD}^{\mathrm{t}}(\cdot)$, introduced by Sipser [Sip83]. We say that a description (called a *program* in the Kolmogorov complexity literature) *p distinguishes* a string x if p accepts x and only x. $\mathrm{CD}^{\mathrm{t,A}}(\mathrm{x})$ is the size of the smallest program that distinguishes x and that runs in time $t(|x|)$ with access to the oracle A. Sipser showed that, for every set B and every length n, there is a string w of length $\mathrm{poly(n)}$ such that, for every $x \in B^{=n}$, $\mathrm{CD}^{\mathrm{poly,B^{=n}}}(\mathrm{x} \mid \mathrm{w}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \log\log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(1)$. Hence the optimal upper bound of $\log(|B^{=n}|)$ can be achieved (almost) if we allow the distinguishing program to use an advice of polynomial length. Later, Buhrman, Fortnow, and Laplante [BFL01] showed how to avoid the advice at the expense of the length of compression. More precisely, they showed that for some polynomial p, for every set B, and every string $x \in B^{=n}$, $\mathrm{CD}^{\mathrm{p,B^{=n}}}(\mathrm{x}) \leq 2\log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}(\log \mathrm{n})$. Hence a significant question is whether the factor 2 in the length of compression is necessary if no advice is given. Interestingly, Buhrman, Laplante, and Miltersen [BLM00] showed that in the general setting, there are sets for which the answer is yes. In particular, they showed that there is a set B such that for sufficiently large n, there exists $x \in B^{=n}$ for which $\mathrm{CD}^{\mathrm{t,A}} \geq 2\log|\mathrm{B}^{=\mathrm{n}}| - \mathrm{O}(1)$ for a t that is super-polynomial.

There are results in the literature where the upper bound is $\log(|B^{=n}|)$ ($+$ a small "precision" term) at the price of weakening other parameters. For example, Buhrman, Fortnow, and Laplante [BFL01] showed such an upper bound of $\log(|B^{=n}|)$ in an average-case setting: For any B, any ε, for all except a fraction of ε strings $x \in B^{=n}$, $\mathrm{CD}^{\mathrm{poly,B^{=n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{poly}\log(\mathrm{n} \cdot 1/\varepsilon)$. The precision term $\mathrm{poly}\log(\mathrm{n} \cdot 1/\varepsilon)$ has been improved to $O(\log(n \cdot 1/\varepsilon))$ in [BMVZ13]. In addition, Buhrman, Lee, and van Melkebeek [BLvM05] showed that for all B and $x \in B^{=n}$, $\mathrm{CND}^{\mathrm{poly,B^{=n}}}(\mathrm{x}) \leq \log(|\mathrm{B}^{=\mathrm{n}}|) + \mathrm{O}((\sqrt{\log(|\mathrm{B}^{=\mathrm{n}}|)} + \log \mathrm{n})\log \mathrm{n})$, where CND is similar to CD except that the distinguisher program is nondeterministic.

Our Contribution

In this paper we consider the following question. Is it possible to achieve the optimal $\log|B^{=n}|$ bound for the language compression problem in the case where we have a bound on the complexity of the language we are compressing? Our main result is that, under a certain reasonable hardness assumption that is used in the area of derandomization, the upper bound of $\log(|B^{=n}|)$ holds for every set B in the class PSPACE/poly. We state the result below. For a precise and stronger statement, see Theorem 3.1.

Main Result. Assume that there exists $f \in \mathrm{E}$ *that cannot be computed in space* $2^{o(n)}$*. Then for any A in* PSPACE/poly*, there exists a polynomial p such that for every* $x \in A^{=n}$*,*

$$\mathrm{CD}^{p,A^{=n}}(x) \leq \log(|A^{=n}|) + O(\log n).$$

The main result is a corollary of the following stronger result: Under the same hardness assumption, the distinguisher program p for x of length $\log(|A^{=n}|) + O(\log n)$ is simple conditioned by x, in the sense that $C^{\mathrm{poly}}(p \mid x) = O(\log n)$, where $C^{\mathrm{poly}}(\cdot)$ is the polynomial-time bounded Kolmogorov complexity function. The idea of the proof is that the hardness assumption implies efficient pseudo-random generators that are sufficient to *derandomize* Sipser's non-uniform result (which is a consequence of a randomized construction).

We also show that in a natural sense the main result is optimal by showing that for any super-polynomial function t that is time constructible, there is a set A computable in space $t(n)$ for which the lower bound from [BLM00] holds: For every polynomial p, for every sufficiently large n, there exists $x \in A^{=n}$ with $\mathrm{CD}^{p,A^{=n}}(x) \geq 2\log(|A^{=n}|) - O(1)$.

Finally we consider the applicability of pseudo-random generator constructions for the problem of (efficiently) compressing/decompressing efficient languages, originally considered by Goldberg and Sipser [GS85] and subsequently treated by Trevisan, Vadhan, and Zuckerman [TVZ05]. We show that the hardness assumption that there exists $f \in \mathrm{E}$ that cannot be computed in space $2^{o(n)}$, leads to improved results on this compression problem (see Section 5 for precise statement of results.)

It is natural to investigate to what extent basic results in Kolmogorov complexity remain valid in resource-bounded Kolmogorov complexity, especially for the case of polynomial bounds. In general, proofs in Kolmogorov complexity can be easily adapted for the *space-bounded* version. The case of *time-bounded* Kolmogorov complexity is quite different. Most proofs cannot be converted from the classical setting to the polynomial-time setting and, in fact, some results no longer hold. For example, symmetry of information does not hold for polynomial-time-bounded Kolmogorov complexity [LW95] (provided one-way permutations exist, which is generally believed to be true).

Recently, certain techniques based on the theory of pseudo-randomness in computational complexity have been used to obtain interesting results in polynomial-time-bounded Kolmogorov complexity. For instance, Antunes and Fortnow [AF09] have shown that, under a certain reasonable hardness assumption, $2^{-C^p(x)}$ dominates all P-samplable distributions (where $C^p(x)$ is the length of the shortest program that generates x within time $p(|x|)$). Our approach uses the similar machinery and reinforces the fact that this is a powerful technique in time-bounded Kolmogorov complexity research.

2 Preliminaries

We use $|A|$ to denote the size of a finite set A, $[M]$ to denote the finite set $\{1,\ldots,M\}$, and $\log(.)$ to denote $\log_2(.)$.

We use standard notation and concepts from computational complexity and Kolmogorov complexity (see [AB09], [LV08]). We are interested in languages in PSPACE/poly. A language A is in PSPACE/poly if there exist a machine M and a polynomial p such that for all input lengths n there is a string y of length $p(n)$, and for all x of length n, $x \in A$ if and only if $M(x,y)$ accepts x in $p(n)$ space. The advice represents information that is nonuniform across input lengths, and is given to the machine along with the input.

$C^t(x)$ denotes the t-time-bounded Kolmogorov complexity of x. Formally, $C^t(x)$ is the minimal length of a program p such that a fixed universal machine U on input p prints x in at most $t(|x|)$ steps. Since a different universal machine affects the Kolmogorov complexity only by an $O(1)$ additive term, and the time bound by a $O(\log t)$ multiplicative factor, in the rest of the paper we will fix one arbitrary universal machine and work with it.

Our main focus in this paper is a variation of time-bounded Kolmogorov complexity called *distinguishing complexity* introduced by Sipser [Sip83]. The t-time bounded distinguishing complexity of x, denoted $\mathrm{CD^t(x)}$ is the length of the shortest program p that accepts x and only x within time $t(|x|)$. We define this formally next.

Definition 2.1 (Distinguishing complexity)**.** *The t-time bounded distinguishing complexity of x, denoted* $\mathrm{CD^t(x)}$, *is the length of the shortest program p such that (1) $U(p,x)$ accepts, (2) $U(p,v)$ rejects for all $v \neq x$, and (3) $U(p,v)$ halts in at most $t(|v|)$ steps for all v.*

Here U is the type of universal Turing machine typically used for time-bounded Kolmogorov complexity. If U is an oracle machine, we define in a similar way $\mathrm{CD^{t,A}(x)}$, by allowing U to query the oracle A. We fix U and we call a string p as above, a program. We use the notation $p(x)$ as a substitute for $U(p,x)$ and $p^A(x)$ as a substitute for $U^A(p,x)$ (i.e., A is the oracle used by program p).

Our main tool is hardness based pseudo-random generators which we discuss next.

Hardness assumptions and pseudo-random generators

The proof of the main result relies on probability distributions that (1) have small support, (2) are efficiently samplable, and (3) cannot be distinguished from the uniform distribution by certain predicates. More precisely, we need pseudo-random generators that extend a seed of length $O(\log n)$ to a string of length n in time polynomial in n, and such that the output “looks” uniformly random to certain predicates T of bounded complexity. Formally, a pseudo-random generator $g : \{0,1\}^{c\log n} \to \{0,1\}^n$ fools a predicate T if

$$\mathrm{|Prob_{z\in\{0,1\}^{c\log n}}[T(g(s))=1]-Prob_{z\in\{0,1\}^n}[T(u)=1]| < 1/n.}$$

By the celebrated result of Impagliazzo and Wigderson [IW97], strengthening an earlier result of Nisan and Wigderson [NW94], and its relativization obtained by Klivans and van Melkebeek [KvM02], it follows that certain hardness assumptions imply the existence of pseudo-random generators of the type that we need. Let $f : \{0,1\}^* \to \{0,1\}$ be some function and $T \subseteq \{0,1\}^*$ be a set (viewed also as a predicate via the identification with its characteristic function). Let $S_f^T(n)$ denote the size of the smallest circuit with T gates that computes the function f for inputs of length n. For $\mathscr{C}$ a complexity class (such as PSPACE, NP, or Σ_p^k, the k-th level of the polynomial hierarchy), we use $S_f^{\mathscr{C}}(n)$ to denote $S_f^T(n)$ for some predicate T that is complete under polynomial-time reduction for the class $\mathscr{C}$. We denote $\mathrm{E} = \cup_{c>0}\mathrm{DTIME}[2^{cn}]$.

Assumption $H(T)$: There exists a function f in E such that, for some $\varepsilon > 0$, $S_f^T(n) > 2^{\varepsilon n}$.

Theorem 2.2 (Klivans and van Melkebeek [KvM02]). *If $H(T)$ is true, then there exists a constant c and a pseudo-random generator $g : \{0,1\}^{c\log n} \to \{0,1\}^n$ that fools the predicate T and such that, for every $s \in \{0,1\}^{c\log n}$, $g(s)$ is computable in time polynomial in n.*

In our main application the set T is in PSPACE/poly. For such T, one can use the following hardness assumption H_1 that is less technical and is still plausible.

Assumption H_1: There exists a function f in E such that, for some $\varepsilon > 0$, $S_f^{\mathrm{PSPACE}}(n) > 2^{\varepsilon n}$.

The following lemma is easy to prove.

Lemma 2.3. *If $T \in$ PSPACE/poly, then H_1 implies $H(T)$.*

One can also use a hardness assumption that only involves uniform computation which is more cleaner to state (i.e., no circuits or advice information).

Assumption H_2: There exists a function f in E which is not computable in space $2^{o(n)}$.

More explictly, this means that f is in E, and for every machine M that computes f there exists a constant $\varepsilon > 0$ such that, for all sufficiently large n, M requires space at least $2^{\varepsilon n}$, on some input of length n.

Lemma 2.4 (Miltersen [Mil01]). *H_2 implies H_1.*

3 Main result

In this section we state and prove our main theorem.

Theorem 3.1. *Assume H_1 holds. Then for every A in* PSPACE/poly, *there exists a polynomial p such that, for all $x \in A^{=n}$, $\mathrm{CD}^{p,A^{=n}}(x) \le \log|A^{=n}| + O(\log n)$.*

Proof. Let A in PSPACE/poly. Fix n, and let $k = \lceil \log|A^{=n}| \rceil$. Let $\mathscr{H}$ be the set of linear functions $h : \{0,1\}^n \to \{0,1\}^{k+1}$. Each $h \in \mathscr{H}$ is given by a $(k+1)\times n$ matrix H over GF[2] and $h(x) = Hx$. We say that "h isolates x" if for all $y \in A^{=n} \setminus \{x\}$, $h(x) \neq h(y)$. It is easy to check that, for fixed x and y in $A^{=n}$, $\mathrm{Prob}_{h\in\mathscr{H}}[h(x) = h(y)] = 1/2^{k+1}$. Thus, for fixed x in $A^{=n}$, $\mathrm{Prob}_{h\in\mathscr{H}}[\text{h does not isolate x}] \le |A| \cdot 1/2^{k+1} \le 1/2$. If we take uniformly at random a tuplet of $(k+1)$ functions $\overline{h} = (h_1, \ldots, h_{k+1}) \in \mathscr{H}^{k+1}$, $\mathrm{Prob}_{\overline{h}}[\text{no } h_i \in \overline{h} \text{ isolates x}] \le 1/2^{k+1}$. Therefore, $\mathrm{Prob}_{\overline{h}}[(\exists x \in A^{=n}) \text{ no } h_i \in \overline{h} \text{ isolates x}] \le |A| \cdot 1/2^{k+1} \le 1/2$.

Note that given $\overline{h} = (h_1, \ldots, h_{k+1})$ such that some $h_i \in \overline{h}$ isolates x, x can be described (in the sense of $\mathrm{CD}^{\mathrm{poly},A}()$) by i and $h_i(x)$, an information which has length $\log k + (k+1) = \log|A^{=n}| + O(\log n)$. The problem is that the length of $\overline{h}$ is $n(k+1)^2$.

We can obtain a shorter such $\overline{h}$ using the assumption H_1 and the pseudorandom generator implied by it. Let $T(\overline{h})$, where $\overline{h} = (h_1, \ldots, h_{k+1}) \in \mathscr{H}^{k+1}$, be the predicate defined by

$$T(\overline{h}) = 1 \text{ iff } (\forall x \in A^{=n})(\exists h_i \in \overline{h}, h_i \text{ isolates } x). \tag{1}$$

Since A is in PSPACE/poly, it is easy to check that the predicate T is computable in PSPACE/poly. Therefore assumption H_1 implies assumption $H(T)$, which, at its turn, implies the existence of a pseudo-random generator $g : \{0,1\}^{c \log n} \to \{0,1\}^{n(k+1)^2}$ such that

$$\mathrm{Prob}_{s \in \{0,1\}^{c \log n}}[T(g(s)) = 1] \geq \mathrm{Prob}_{\overline{h}}[T(\overline{h}) = 1] - 1/n \geq 1/2 - 1/n > 0.$$

Therefore, there exists $s \in \{0,1\}^{c \log n}$ such that $g(s)$ is a tuplet $\overline{h} = (h_1, \ldots, h_{k+1})$ with the property that for every $x \in A$, there is some $i \in \{1, \ldots, k+1\}$ such that h_i isolates x. Thus, any string x in A can be described by $p = (k, s, i, h_i(x))$, an information which can be encoded with $k + O(\log n)$ bits. Indeed, on input (p, v), the distinguishing algorithm constructs the pseudo-random generator g (which depends on k and $n = |x|$), calculates $g(s) = (h_1, \ldots, h_{k+1})$, and accepts if and only if $h_i(v) = h_i(x)$. By the discussion above, this algorithm only accepts the string x. $\square$

Remark. The proof shows more: The program $p = (k, s, i, h_i(x))$ witnessing that $\mathrm{CD}^{\mathrm{poly}, A^{=n}}(x) \leq \log |A^{=n}| + O(\log n)$, can be obtained from x, k, i, s in polynomial time, and therefore $C^{\mathrm{poly}}(p \mid x) \leq O(\log n)$.

Using the same technique, the following result for sets in the polynomial-time hierarchy with polynomial advice holds (the conclusion is weaker than in Theorem 3.1, but so is the assumption).

Theorem 3.2. *Assume that there exists a function f in E such that, for some $\varepsilon > 0$, $S_f^{\Sigma_k^p}(n) \geq 2^{\varepsilon n}$, where k is a natural number. Then for every A in Σ_k^p/poly, there exists a polynomial p such that, for all $x \in A^{=n}$, $\mathrm{CD}^{p,A}(x) \leq \log |A^{=n}| + O(\log n)$.*

4 A lower bound for sets computable in superpolynomial space

We show that the complexity class PSPACE/poly is essentiallly the largest class for which the optimal language compression in Theorem 3.1 holds. Indeed, if $t(n)$ is a function that is superpolynomial and time-constructible, then there exists a set A computable in space $t(n)$ for which the $2 \log |A^{=n}| - O(1)$ lower bound shown by Buhrman, Laplante, and Miltersen [BLM00] holds.

Theorem 4.1. *Let $t(n)$ be a superpolynomial and time-constructible function. There exists a set A computable in space $t(n)$ such that for all sufficiently large n, there exists $x \in A^{=n}$ such that*

(1) $\mathrm{CD}^{t(n)^{1/2}, A}(x) \geq 2 \log |A^{=n}| - O(1)$,

(2) $|A^{=n}| \geq t(n)^{1/6}$.

The proof follows closely the arguments from [BLM00] (we add only elements that determine the space complexity bound for the set A). For completeness, we present it below. The key part is a combinatorial result on k-cover-free subsets. A k-cover-free family $\mathscr{F}$ is a family of N subsets of $[M]$ with the property that no subset in $\mathscr{F}$ is contained in the union of k other subsets in $\mathscr{F}$. More precisely, if $F_0, F_1, \dots, F_k$ are distinct subsets in $\mathscr{F}$, then $F_0 \not\subseteq F_1 \cup \ldots \cup F_k$. Dyachkov and Rykov [DR82] have shown that if $k \leq N^{1/3}$, then $M \geq \frac{k^2 \log N}{2\log k + c}$, for some constant c.

Theorem 4.2 (Dyachkov and Rykov [DR82])**.** *Let $\mathscr{F}$ be a family of N subsets of $[M]$. Then if $\mathscr{F}$ is k-cover-free for $k \leq N^{1/3}$, then $M \geq \frac{k^2 \log N}{2\log k + c}$, for some constant c.*

Proof of Theorem 4.1. Let us fix n (sufficiently large) and define $r = \min\{\lceil t(n)^{1/2} \rceil, 2^{n/8}\}$, and $k = \lceil r^{1/3} \rceil$. Let $\mathscr{P}$ be the set of programs of length at most $2\log(k+1) - c_1$ (for a constant c_1 that will be specified later), which run in time bounded by r. Clearly, $|\mathscr{P}| \leq (k+1)^2/c_1$.

First we find the lexicographically smallest string y of length $r \cdot n$ such that if $y = x_1 x_2 \ldots x_r$, with every $x_i \in \{0,1\}^n$, then the strings $x_1, \dots, x_r$ are all distinct and $C^{3r}(y) \geq r \cdot n - 1$ (recall that $C^{3r}(y)$ is the Kolmogorov complexity of y with time bounded by $3r$). By a simple counting argument such a string exists and, furthermore, it can be found in space bounded by $r \cdot n + 3r = o(t(n)^{1/2})$. Let $B = \{x_1, \dots, x_r\}$. Note that for any two distinct string $x_i, x_j \in B$, $C^r(x_i \mid x_j) \geq n/2$ (otherwise y could be reconstructed in $3r$ steps from information less than $r \cdot n - 1$).

This implies that for any set $A \subseteq B$, for any program $p \in \mathscr{P}$, and for any $x \in A$,

$$p^{\{x\}}(x) = p^A(x). \tag{2}$$

This is true because otherwise, the program p on input x and with oracle $\{x\}$, during its r-step computation, queries the oracle about some string $u \in A \setminus \{x\}$. Note that $C^r(u \mid x) \leq \log r + |p| + O(1) < n/2$, which contradicts the property of the elements of B.

Claim 4.3. *There exists a set $A \subseteq B$, $|A| = k+1$ and $x \in A$ such that for every $p \in \mathscr{P}$, either*

(1) $p^{\{x\}}(x)$ *does not accept, or*

(2) $p^{\{x\}}(x)$ *accepts and there exists $y \in A \setminus \{x\}$ such that $p^{\{y\}}(y)$ accepts.*

The statement of the theorem is a consequence of Claim 4.3, as can be seen from the following two observations. First, by Equation (2), for every $p \in \mathscr{P}$, either

(1) $p^A(x)$ does not accept, or

(2) $p^A(x)$ accepts and there exists $y \in A \setminus \{x\}$ such that $p^A(y)$ accepts.

This implies that $\mathrm{CD}^{\mathrm{r,A}}(\mathrm{x}) \geq 2\log(\mathrm{k}+1) - \mathrm{O}(1) = 2\log|\mathrm{A}^{=\mathrm{n}}| - \mathrm{O}(1)$. Secondly, it is easy to see that, by an exhaustive search, one can find a set A satisfying the conditions in Fact 4.3 and print out its elements in space $O(r)$, and thus the set A is computable in space $t(n)^{1/2}$.

Proof of Claim 4.3. It remains to show Claim 4.3. To reach a contradiction, suppose that for every subset $A \subseteq B$ of size $k+1$ and for every $x \in A$ there exists $p \in \mathscr{P}$ such that $\{u \in A \mid p^{\{u\}}(u) \text{ accepts }\} = \{x\}$. We define a family of r subsets as follows: For every $x \in B$, let $F_x = \{p \in \mathscr{P} \mid p^{\{x\}}(x) \text{ accepts }\}$. $\{F_x \mid x \in B\}$ is a family of r subsets of $\{1, \ldots, (k+1)^2/c_1\}$ which by our assumption is k-cover-free. By the combinatorial result of Diachkov and Rykov, $\frac{(k+1)^2}{c_1} \geq \frac{k^2 \log r}{2 \log k + c}$, which, for large enough c_1, is a contradiction. □

5 Compression of efficient languages

We show that hardness based pseudo-random generator constructions can be used to get improved upper bound on standard compression/decompression problem of efficient languages first considered by Goldberg and Sipser [GS85] (this is different from the language compression problem given by the $\mathrm{CD}^{\mathrm{poly}}(\cdot)$ complexity). In the standard compression problem there are two steps: (1) given a string x we seek a succinct representation of it, the string y (compression), and (2) given y, we want to reconstruct x (decompression).

Definition 5.1. *A set A is compressible by a uniform family of algorithms $(Enc_n, Dec_n)_{n \geq 1}$ to length $m(n)$ if for each n,*

(1) $Enc_n : \{0,1\}^n \to \{0,1\}^{m(n)}, Dec_n : \{0,1\}^{m(n)} \to \{0,1\}^n$,

(2) For all $x \in A^{=n}$, $Dec_n(Enc_n(x)) = x$.

Goldberg and Sipser [GS85] showed that any set A in P with $|A^{=n}| \leq 2^n/n^k$, can be compressed in polynomial time to length $n - (k-3)\log n$. Their algorithm is probabilistic and hence can be erroneous with some small probability. Later, Trevisan, Vadhan and Zuckerman [TVZ05] gave, for any set A in P, compression algorithms to length $k(n) + \mathrm{polylog}(n - k(n))$ that run in time $2^{n-k(n)} \cdot \mathrm{poly(n)} \cdot 2^{\mathrm{poly}\log(1/\varepsilon)}$, where $k(n) \geq \lceil \log |A^{=n}| \rceil$ and $k(n)$ is computable in time poly(n). However, their algorithms only work for a $(1-\varepsilon)$ fraction of strings in $A^{=n}$. We show that, under the hardness assumption H_1, compression/decompression can be done for all strings in a language in P with parameters essentially identical to the above results, but without any error.

Theorem 5.2. *Assume H_1. Let A be a set in P such that for every n, $|A^{=n}| \leq 2^n/n^2$. Let $k(n) \geq \lceil \log |A^{=n}| \rceil$ be a function computable in polynomial time on input 1^n. Then A is compressible to length $k(n) + O(\log n)$ by algorithms (Enc_n, Dec_n) running in time $2^{n-k(n)} \cdot$ poly(n).*

Remarks: We make two remarks about this theorem. (1) If $k(n) = n - O(\log n)$, (Enc_n, Dec_n) run in polynomial time, are deterministic, and work for all strings in $A^{=n}$. (2) The assumption H_1 can be replaced by the (probably) weaker assumption: There exists f in E such that, for some $\varepsilon > 0$, $S_f^{\mathrm{NP}} > 2^{\varepsilon n}$.

Proof. The proof is a variation of the proof of Theorem 3.1, and we use the notation from that proof. We fix length n and let $k = k(n)$. We define the predicate $\tilde{T}$ by adding to the predicate T from equation (1), the condition that all the linear functions h_i have rank $(k+1)$. Thus,

$$\tilde{T}(\overline{h}) = 1) \text{ iff } T(\overline{h}) = 1 \text{ and } (\forall h_i \in \overline{h}, h_i \text{ has rank } k+1)].$$

The probability, when $\overline{h}$ is chosen at random in $\mathscr{H}^{k+1}$, of the event "$\tilde{T}(\overline{h}) = 1$" is at least $1/3$, because the probability that a random $(k+1) \times n$ matrix has rank less than $k+1$ is approximately $O(2^{-(n-k)})$, and therefore the probability of the event "$\tilde{T}(\overline{h}) = 1$" is only slightly less than the probability of the event "$T(\overline{h}) = 1$," which we have seen to be at least $1/2$. Consequently, there exists a pseudo-random generator $g_k : \{0,1\}^{c \log n} \to \{0,1\}^{n(k+1)^2}$ such that

$$\mathrm{Prob}_{\mathrm{s} \in \{0,1\}^{\mathrm{c} \log \mathrm{n}}}[\tilde{\mathrm{T}}(\mathrm{g_k}(\mathrm{s})) = 1] > 0.$$

The compression algorithm Enc_n on input $x \in A^{=n}$ works as follows: It finds a seed s such that if $g_k(s) = (h_1, \ldots, h_{k+1})$, there exists h_i that isolates x and such that h_i has rank $(k+1)$. This takes time $2^{n-k} \cdot \mathrm{poly(n)}$ because if h_i has rank $(k+1)$, there are at most 2^{n-k-1} strings x' such that $h_i(x') = h_i(x)$. Then, the algorithm Enc_n on input x returns $(k, n, s, i, h_i(x))$.

The decompression algorithm Dec_n on input (k, n, s, i, y), constructs g_k, then $g_k(s) = (h_1, \ldots, h_{k+1})$ and determines the set $h_i^{-1}(y)$ which has at most 2^{n-k-1} elements. Then it searches for the unique element in that set that belongs to $A^{=n}$, and returns that element.

□

References

[AB09] S. Arora and B. Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[AF09] L. Antunes and L. Fortnow. Worst-case running times for average-case algorithms. In *Proceedings of the 24th Conference in Computational Complexity Conference*, pages 598–303. IEEE Computer Society Press, 2009.

[BFL01] H. Buhrman, L. Fortnow, and S. Laplante. Resource-bounded Kolmogorov complexity revisited. *SIAM J. Comput.*, 31(3):887–905, 2001.

[BLM00] H. Buhrman, S. Laplante, and P.B. Miltersen. New bounds for the language compression problem. In *IEEE Conference on Computational Complexity*, pages 126–130, 2000.

[BLvM05] H. Buhrman, T. Lee, and D. van Melkebeek. Language compression and pseudorandom generators. *Computational Complexity*, 14(3):228–255, 2005.

[BMVZ13] B. Bauwens, A. Makhlin, N. Vereshchagin, and M. Zimand. Short lists with short programs in short time. *ECCC*, TR13-007, 2013.

[DR82] A. G. Dyachkov and V. V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13, 10982. In Russian.

[GS85] A. Goldberg and M. Sipser. Compression and ranking. In *Proceedings of the 17th ACM Symposium on Theory of Computing*, pages 440–448, 1985.

[IW97] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 220–229, New York, May 1997. Association for Computing Machinery.

[KvM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

[LR05] T. Lee and A. E. Romashchenko. Resource bounded symmetry of information revisited. *Theor. Comput. Sci.*, 345(2-3):386–405, 2005.

[LV08] M. Li and P. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag, 2008. 3rd edition. 1st edition in 1993.

[LW95] L. Longpré and O. Watanabe. On symmetry of information and polynomial time invertibility. *Inf. Comput.*, 121(1):14–22, 1995.

[Mil01] P. B. Miltersen. Derandomizing complexity classes. In P. Pardalos, J. Reif, and J.Rolim, editors, *Handbook on Randomized Computing, Volume II*. Kluwer Academic Publishers, 2001.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 330–335, 1983.

[TVZ05] L. Trevisan, S. P. Vadhan, and D. Zuckerman. Compression of samplable sources. *Computational Complexity*, 14(3):186–227, 2005.